CLAIMS

What is claimed is:

1. A system for generating a unified user profile to allow transparent access to multiple data sources, the system comprising:

(a) a first data source;

(b) a second data source; and

(c) a server adapted to access said first and second data source, said server comprising a component adapted to aggregate data from said first and second data sources into a unified user profile.

2. A system according to claim 1, wherein said first data source is selected from the group consisting of legacy databases, corporate databases, and user data stores.

3. A system according to claim 1, wherein said first data source contains data selected from the group consisting of authentication information, user lists, group lists, and group membership.

4. A system according to claim 1, further comprising a security realm adapted to allow authentication of data in at least one of said first and second data sources.

5. A system according to claim 1, wherein said server is a personalization server.

6. A system according to claim 1, wherein said component comprises an enterprise java bean.

7. A system according to claim 6, wherein said enterprise java bean retrieves and updates data in at least one of said first and second data sources using methods selected from the group consisting of getProperty() and setProperty().

8. A system according to claim 1, wherein said component comprises an extended java bean.

9. A system according to claim 1, wherein said component provides a transparent interface through which implicit and explicit properties can be retrieved and updated.

10. A system according to claim 9, wherein said component comprises a property set, said property set adapted to give namespace qualifications to said implicit and explicit properties.

11. A system according to claim 1, wherein said component comprises getter and setter properties.

12. A system according to claim 1, wherein said component provides a transparent interface adapted to store and retrieve data from said first data store and said second data store.

13. A system according to claim 1, wherein said second data source is a personalization database.

14. An architecture for generating a unified user profile for transparent access to existing user data, the architecture comprising:

(a) a base user enterprise Java bean, said base user enterprise Java bean capable of being extended to incorporate said existing user data;

(b) a user data store adapted to contain said existing user data; and

(c) a user-specific enterprise java bean, adapted to provide transparent read and write access to said existing user data.

15. An architecture according to claim 14, further comprising a data source containing data external to said existing user data.

16. An architecture according to claim 15, wherein said user-specific enterprise Java bean further allows transparent read and write access to said data in said data source.

17. An architecture according to claim 14, further comprising a server adapted to provide said read and write access to a user of said unified user profile.

18. An architecture according to claim 17, wherein said server is a personalization server.

19. An architecture according to claim 14, wherein said user data store is a table in an internal data source selected from the group consisting of legacy databases, corporate databases, and customer databases.

20. An architecture according to claim 14, wherein said user data store contains data selected from the group consisting of authentication information, user lists, group lists, and group membership.

21. An architecture according to claim 14, further comprising a security realm adapted to allow authentication of data in said user data store.

22. An architecture according to claim 14, wherein said user-specific enterprise Java bean utilizes a property set, said property set adapted to give namespace qualifications to implicit and explicit properties of said existing user data.

23. An architecture according to claim 14, wherein said user-specific enterprise Java bean utilizes getter and setter properties.

24. A method for generating a unified user profile for providing transparent access to a personalization database and external user database, said method comprising the steps of:

(a) obtaining a base user java bean adapted to work through a personalization server to access said personalization database, said base user java bean adapted to provide a transparent interface through which implicit and explicit properties can be retrieved and updated from the personalization database; and

(b) creating an enterprise java bean to extend the base user java bean such that said implicit and explicit properties can further be retrieved and updated from an external user database.

25. A method according to claim 24, further comprising the step of generating transparent read and write access to said external database through the extended said base user java bean.

26. A method according to claim 24, further comprising the step of configuring a server to provide said read and write access.

27. A method according to claim 26, wherein said server is a personalization server.

28. A method according to claim 24, wherein said external user database is selected from the group consisting of legacy databases, corporate databases, and customer databases.

29. A method according to claim 24, wherein said external user database contains data selected from the group consisting of authentication information, user lists, group lists, and group membership.

30. A method according to claim 24, further comprising the step of obtaining a security realm adapted to allow authentication of data in said personalization database and said external user database.

31. A method according to claim 24, wherein the extended base user java bean utilizes a property set, said property set adapted to give namespace qualifications to implicit and explicit properties of said data in said personalization database.

32. A method according to claim 31, wherein said implicit and explicit properties comprise getter and setter properties.

33. A method for transparently accessing multiple data sources, said method comprising the steps of:

(a) obtaining a base user java bean adapted to work through a server to access an internal data source, said base user java bean adapted to provide a transparent interface through which implicit and explicit properties can be retrieved and updated; and

(b) extending the user java bean such that said base user java bean is further adapted to provide a transparent interface through which implicit and explicit properties can be retrieved and updated from at least one external data source.

34. A method according to claim 33, further comprising the step of configuring a server to operate said transparent interface.

35. A method according to claim 33, further comprising the step of obtaining a security realm adapted to allow authentication of data in said internal data source and said external data source.

36. An method according to claim 33, further comprising the step of configuring a property set for the extended user java bean.

37. A method according to claim 35, wherein said property set is adapted to give namespace qualifications to implicit and explicit properties of said data in said internal and external data sources.

38. A method according to claim 37, wherein said implicit and explicit properties comprise getter and setter properties.

39. A method according to claim 37, further comprising the step of using reflection to determine whether a property of said data in said internal and external data sources is explicit.

40. A system for transparently accessing multiple data sources, said system comprising:

(a) a plurality of data sources;

(b) a server in communication with each said data source; and

(c) an extended user java bean adapted to provide transparent access to said plurality of data sources through said server.

41. A system according to claim 40, wherein at least one of said plurality of data sources is selected from the group consisting of legacy databases, corporate databases, and user data stores.

42. A system according to claim 40, further comprising a security realm adapted to allow authentication of data in at least one of said plurality of data sources.

43. A system according to claim 40, wherein said server is a personalization server.

44. A system according to claim 40, wherein said extended user java

bean retrieves and updates data in at least one of said plurality of data sources using methods selected from the group consisting of getProperty() and setProperty().

45. A system according to claim 40, wherein said extended user java bean is adapted to allow implicit and explicit properties of data in said plurality of data sources to be retrieved and updated.

46. A system according to claim 45, wherein said extended user java bean utilizes a property set, said property set adapted to give namespace qualifications to said implicit and explicit properties.

47. A system according to claim 45, wherein said implicit and explicit properties comprise getter and setter properties.

48. A system for unifying multiple data sources, said system comprising:

(a) a naming convention to be followed in storing and accessing data in the data sources;

(b) a plurality of data sources, at least one data source containing a data entry not following said naming convention;

(c) a set of identifier pairs, each identifier pair corresponding to a data entry that does not follow said naming convention, the identifier pair including the name of the entry and a corresponding name that follows the naming convention; and

(d) a server in communication with each data source and the set of identifier pairs, the server adapted to allow access to the data sources by a request following said naming convention.

49. A system according to claim 48, wherein at least one of said plurality of data sources is selected from the group consisting of legacy databases, corporate databases, and user data stores.

50. A system according to claim 48, further comprising a security realm adapted to allow authentication of data in at least one of said plurality of data sources.

51. A system for generating a unified user profile adapted to allow transparent access to multiple data sources, the system comprising a server including:

(a) a first component adapted to access a first data source;

(b) a second component adapted to access a second data source; and

(c) a user component adapted to aggregate data from the first and second data sources into a unified user profile.

52. A system according to claim 51, further comprising component adapted to access a security realm for authentication of data in at least one of said first and second data sources.

53. A system according to claim 51, wherein the user component comprises an enterprise java bean.

54. A system according to claim 51, wherein the user component retrieves and updates data in at least one of the first and second data sources using methods selected from the group consisting of getProperty() and setProperty().

55. A system according to claim 51, wherein the user component provides a transparent interface through which implicit and explicit properties can be retrieved and updated.

56. A system according to claim 55, wherein the user component comprises a property set, said property set adapted to give namespace qualifications to said implicit and explicit properties.

57. A system according to claim 51, wherein the user component comprises getter and setter properties.

58. An architecture for generating a profile adapted to provide access to user data, the architecture comprising:

(a) a base user enterprise Java bean, said base user enterprise Java bean capable of incorporating the user data; and

(b) a user-specific enterprise java bean, adapted to provide transparent read and write access to the user data.

59. An architecture according to claim 58, further comprising a server adapted to provide the read and write access to the user data.

60. An architecture according to claim 58, wherein said user data store contains data selected from the group consisting of authentication information, user lists, group lists, and group membership.

61. An architecture according to claim 58, wherein said user-specific

enterprise Java bean utilizes a property set, said property set adapted to give namespace qualifications to implicit and explicit properties of the user data.

62. An architecture according to claim 59, wherein the user-specific enterprise Java bean utilizes getter and setter properties.

63. A computer readable medium containing instructions which, when executed by a server, cause the server to perform the steps of:

(a) obtaining a base user java bean adapted to work through the server to access a first database, said base user java bean adapted to provide a transparent interface through which implicit and explicit properties can be retrieved and updated from the first database; and

(b) creating an enterprise java bean to extend the base user java bean such that said implicit and explicit properties can further be retrieved and updated from a second database.

64. A computer readable medium according to claim 63, wherein the medium further causes the server to generate transparent read and write access to the second database through the extended said base user java bean.

65. A computer readable medium according to claim 63, wherein the medium further causes the server to obtain a security realm adapted to allow authentication of data in the first database and the second database.

66. A computer readable medium according to claim 63, wherein the extended base user java bean utilizes a property set, said property set

adapted to give namespace qualifications to implicit and explicit properties of said data in the first database.

67. A computer readable medium according to claim 63, wherein the extended base user java bean utilizes getter and setter properties.